



04

Mapeando una entidad #1

Transcripción

[00:00] Hola y continuando con el desarrollo de nuestro curso. En esta parte nosotros vamos a comenzar a elaborar el modelo que vamos a persistir en nuestra base de datos, entonces nosotros fuimos a tener en nuestra base de datos una tabla que va a representar en nuestro listado de productos y dentro de esa tabla van a existir unas ciertas propiedades como el Id, el nombre, la descripción y el precio.

[00:25] Ese id va a ser el tipo bigint. El nombre va a ser del tipo varchar, la descripción también va a ser del tipo varchar, porque ambos son textos y el precio va a ser del tipo decimal. Entonces nosotros tenemos que darle una interpretación en el código para modelar esa tabla que se encuentra en la base de datos, modelarla en nuestra API, en el proyecto.

[00:52] Entonces para eso, nosotros vamos a construir una clase y vamos a ver cómo Hibernate realiza un mapeamiento o cómo hace la búsqueda de cada uno de los elementos para transformarlos en elementos de la base de datos. Para comenzar este curso, lo primero que tenemos que hacer es crear nuestra clase, para eso vamos a ir a la carpeta principal donde vamos a hacer clic derecho, seleccionar nuevo y vamos a crear una nueva clase.

[01:22] El nombre de esa clase se va a llamar producto, tiene que ser un nombre que tenga lógica y con lo que nosotros estamos haciendo en la base de datos. Nosotros tenemos un listado de productos. Nosotros vamos a guardar esta primera clase en un paquete que va a ser `com.latam.alura.tienda`.

[01:44] Como es parte del modelo con el que vamos a representar nuestra base de datos, vamos a colocar esto dentro de la carpeta modelo. Seleccionamos finalizar y ya con esto tenemos nuestra clase creada. Entonces vemos que creó un paquete que siempre va a estar dentro de la clase modelo y creó la clase que sería la clase producto.

[02:18] Entonces Hibernate él reconoce, lo que serían las entidades, entonces él va a buscar las diversas entidades y esas entidades están representadas por clase en nuestro programa. Entonces, inicialmente de este modo, él nos va a reconocer esa clase, para nosotros indicarle que esta va a ser una de las entidades que debe realizar el mapeamiento, nosotros tenemos las anotaciones.

[02:35] Entonces una de las anotaciones es la anotación `@Entity`. Si nosotros pasamos el mouse acá, vemos que él puede importar esa anotación de `Java-persistence` o de `hibernate-annotation`. Entonces, como nosotros queremos permanecer ágiles y dinámicos, nosotros tenemos que mantenernos con la especificación y no con la implementación.

[03:04] Si nosotros nos mantenemos con la implementación que sería Hibernate a la hora de cambiar Hibernate por alguna otra implementación, por algún otro framework, sería muy difícil, ya que tendríamos que cambiar varios elementos dentro de nuestro proyecto. Entonces nosotros, vamos a seleccionar `java-persistence`.

[03:22] Y para la mayoría de las cosas vamos a estar seleccionando JPA, que es `Java-persistence` para, como habíamos mencionado, mantenernos dentro de la especificación.

[03:33] Entonces, como habíamos mencionado antes, Hibernate y JPA, ellos realizan un mapeamiento de los elementos existentes en nuestra clase para compararlos con el modelo en la base de datos, entonces por default Hibernate

entiende y JPA, entiende que el nombre de la clase es el mismo nombre que existe para la tabla.

[03:58] En nuestro caso no lo es. El nombre de esta clase es la clase producto y la clase en la base de datos es productos en plural. Entonces para nosotros es indicar esa diferencia vamos a usar otra anotación que va a ser `@Table`. Y esa anotación, va a pedir un parámetro a ser `@Table(name="productos")`.

[04:25] Vamos a realizar la importación de JPA también como hemos mencionado y ya con eso nosotros tenemos nuestra clase, vamos a guardarlo, nuestro modelo inicial de entidad elaborado. Ahora está faltando crear los diferentes atributos que van a representar cada una de las columnas en la base de datos.

[04:47] Entonces nosotros teníamos cuatro columnas, era el id, el nombre, la descripción y el precio, entonces vamos a agregar esos elementos. Tenemos un elemento de tipo privado, va a ser del tipo long y va a ser el id. Luego tenemos el nombre que también va a ser de tipo privado, el string. Vamos a tener la descripción, también va a ser un string del tipo privado y por último, vamos a tener un precio.

[05:25] Más adelante nosotros vamos a ir hoy a ver más atributos. Pero por ahora nos vamos a mantener con estos elementos. Y aquí el precio. Aquí voy a importar la clase `BigDecimal`.

[05:48] Y ya con esto tengo los cuatro primeros atributos y el mapeamiento de la entidad. Entonces ahora les voy a mostrar cómo identificar cuál va a ser el id que va a representar cada una de las filas en la base de datos y qué pasa con la manera de generar ese identificador, ese id.